DOCUMENT RESUME

ED 458 099 SE 065 194

AUTHOR Bricker, Lauren J.; Tanimoto, Steven L.; Rothenberg, Alex I.; Hutama, Danny C.; Wong, Tina H.

TITLE Multiplayer Activities That Develop Mathematical Coordination.

PUB DATE 1995-09-00
NOTE 9p.
PUB TYPE Reports - Research (143)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Software; *Computer Uses in Education; *Cooperative Learning; *Educational Technology; Elementary Secondary Education; *Mathematics Education
IDENTIFIERS *Student Cooperation

ABSTRACT

Four computer applications are presented that encourage students to develop "mathematical coordination"--the ability to manipulate numerical variables in cooperation with other students so as to achieve a definite goal. The programs enable a form of computer-supported cooperative learning (CSCL). This paper describes the rationale and design of the programs, the results of an informal evaluation, and possible future work. The games were developed using a special software and hardware environment that facilitates the rapid prototyping of computer-based cooperative-learning materials. This research is part of an ongoing project entitled "Mathematics Experiences through Image Processing," which is designed to develop and test educational materials that introduce K-12 students to mathematical ideas within the context of digital image processing activities. (Contains 19 references.) (MM)

Reproductions supplied by EDRS are the best that can be made
from the original document.

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY
L. Bricker
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

Multiplayer Activities that Develop Mathematical Coordination

Lauren J. Bricker, Steven L. Tanimoto, Alex I. Rothenberg, Danny C. Hutama, and Tina H. Wong

Department of Computer Science and Engineering, University of Washington

ED 458 099

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Abstract

Four computer applications are presented that encourage students to develop "mathematical coordination" -- the ability to manipulate numerical variables in cooperation with other students so as to achieve a definite goal. The programs enable a form of computer-supported cooperative learning (CSCL). In this paper we describe the rationale and design of the programs, the results of an informal evaluation, and possible future work. The games were developed using a special software and hardware environment that facilitates the rapid prototyping of computer-based cooperative-learning materials. This research is part of an ongoing project entitled "Mathematics Experiences Through Image Processing" whose objective is to develop and test educational materials that introduce K-12 students to mathematical ideas within the context of digital image processing activities.

Keywords -- Co-present CSCW, computer supported collaborative learning, mathematics education, mathematical coordination, multiple mice, color matching, curve fitting, chord matching, cooperative learning.

1. Introduction

This paper presents four activities that enable co-present or face-to-face collaborative learning. Computer Supported Collaborative Learning (CSCL) as a whole is designed to encourage cooperation, discussion of ideas, resolution of cognitive conflicts, and promote problem solving and higher-order thinking skills. CSCL programs can be divided into two types: those supporting cooperation at a distance and those supporting cooperation in a co-present setting. As video teleconferencing technology improves and gets cheaper, this dichotomy will become less clear. For now, however, cooperation at a distance is hampered by the low degree of apparent presence available and the lack of actual contact with other students. Co-present collaboration, on the other hand, allows students to interact directly, see each other's expressions and gestures, therefore communicate more effectively.

For large co-present groups (e.g., a classroom full of students), collaborative applications may run on a set of computers interconnected in a local area network. Alternatively, one or more computers may support the users in a "meeting room" configuration. Occasionally one computer is used by a small group of students sharing it simultaneously. While sharing of a single computer may present some logistical problems to the students, the necessity of sharing can promote communication amongst the students. Physically separating students to work individually on computers tends to discourage communication.

The programs presented in this paper were specifically designed to be used in this last situation, that is, by groups of 2-4 students sitting at one machine. The software and hardware allow all the students in the group to interact with the computer simultaneously, via multiple mice and a shared screen. Simultaneous input tends to reduce the conflict over access to the computer, and it may increase the average rate of learning for the students in the group.

The reasons to support cooperative learning has been well stated by Davidson [4] in relation to mathematics. Group learning addresses some of the problems associated with the isolative nature of typical mathematics curricula. As a whole, students working in a group become less discouraged and frustrated than students working alone. The group not only is a source for additional help, but it becomes a support network for members. Combining computers with group learning is an attractive possibility for mathematics education, because computers can empower students to construct and explore mathematical objects and worlds.

The programs we present are intended to support mathematics teaching with the 1989 standards by the National Council of Teachers of Mathematics. Numerous reports have documented the difficulties that K-12 students in the United States have with mathematics (e.g., see [14]). When pressed for a reason, students often complain that mathematics is "difficult" and that they don't see much use for it beyond simple arithmetic. In response to these concerns, the NCTM developed *Curriculum and Evaluation Standards for School Mathematics* ; this document specifies not only the content to be covered but the many ways in which this knowledge should be brought to life and connected to other subjects. These ways include communication about mathematics, problem solving and posing, and integrating the teaching of mathematics with other subjects.

The goal of the *Mathematics Experiences Through Image Processing (METIP)* project is to use digital image processing and collaboration to help meet the NCTM objectives. METIP software and activities allow students to manipulate and view digitized images as objects which are simultaneously visual and mathematical. Students may work assigned problems or explore and make

BEST COPY AVAILABLE

SE 057 914

their own discoveries. Teachers may lead discussions about ideas such as image transformations, invertibility of functions, and effects of arithmetic operations on images that crop up during the computer-based activities.

We have created an environment that facilitates the rapid development of METIP programs. The METIP environment system software is designed in two parts, one which includes image processing primitives and the other, known as the *MultiIn* module, which supports either single-user or co-present CSCL graphical user interfaces. In particular, MultiIn is designed to support a multiplicity of input pointing devices with their tracking cursors all displayed on a single screen. With this system, small groups of students are able to sit at one computer and discuss activities as they are engaged in them.

The next section will discuss the previous work which lead to the decision to support co-present CSCL. In Section 3, we describe four co-present collaborative applications followed by the implementation considerations in Section 5. Section 4 details the preliminary results of our user testing. Finally, Section 6 will include some directions for future work.

2. Previous Work

A number of educational researchers have sought to exploit computer technology in promoting learning through collaboration. For example, Clements used the Logo programming environment in a collaborative setting (i.e. multiple users on a single input machine) to study student learning through cognitive conflict and subsequent resolution [3]. The results of their study suggests that the collaborative use of Logo encourages interaction, discussions, coordination of different ideas and higher order thinking [12]. Unfortunately, the use of the computer in this way also promoted social conflict over the input devices.

TurtleGraph [10] is a similar problem solving environment in which Lisp is used to command the cursor. Additionally the environment has the capability to allow communication between students. The system regulates the communication through button presses and an area for conversational dialog. The restricted communication is designed to make the communication between users more productive.

This type of interaction has been seen in many other distributed CSCL applications [2,11,15,16,17]. Synchronous networked collaboration can be enhanced by adding real time video conferencing, similar to cooperative work systems such as EuroParc's RAVE [8], or Clearboard [9]. Still, because the hardware inhibits direct visual access to partners, these systems do not foster the same level of communication as one finds in face-to-face situations.

Meeting room systems [6, 17, 18, 19] are specifically designed to allow multiple users, each on their own machines, to work collaboratively and still maintain direct face-to-face communication. Some of these systems include special table top monitors that do not obstruct views and/or one large main monitor which is shared by all and serves as a mechanism to synchronize views.

While these systems do encourage communication more than their distributed counterparts do, they do not evoke the excitement and degree of interaction shown by users of co-present video games. Of course, such games are designed to maximize the players excitement and suspense. However it may also be due, to some extent, to limitations of the underlying hardware, such as delays due to network bandwidth, or perhaps it is because the users of such systems are not focusing on exactly the same view of the given problem. Even in systems with a large shared view screen must constantly their attention from their own monitor to the front of the room to gain context.

MMM [1] is a multi-user system in which each person has control of his or her own mouse, but all are working on one screen. This is the same situation found in most interactive home video games developed by Nintendo, Sega and Atari. Ultimately, we would like to channel the excitement and interaction found in video games into an educational context while changing the competition inherent in most games into collaboration.

3. Applications

3.1. Design Goals

We have currently developed four co-present CSCL applications designed to promote strong cooperation between users on a specific activity. Therefore each activity has one definite, predefined goal, and each user must contribute equally towards reaching that goal. In order to successfully complete the task, the users must be able to communicate about the operations on the screen as well as the goal itself. In many cases the users develop their own "language" about the activity.

3.2. The Color Matcher

The Color Matcher is the CSCL application that we developed first. This activity was designed to promote learning and discussion about the RGB additive system used by most monitors for displaying color [7]. It won an industry award for its creative use of the Access.Bus multiple input technology.

The Color Matcher activity gives each of three players access to his or her own mouse which corresponds to a red, green or blue colored cursor on the screen. The goal is for the players to match a target color generated by the computer.

Figure 1 shows the Color Matcher user interface. The interface contains two color frames, the left is the target color and the one on the right is the users' color frame. The users' color is determined by combining the RGB values controlled by three horizontal scroll bars. The color values, which range from 0-255 depending on the position of the slider, are shown above the scroll bars. A scroll bar can be modified only by the cursor of the same color.

Although each user is only in control of his or her own color, the users are collaborating to reach the single goal of matching one color. In order to succeed at this activity, users must, at a minimum, communicate about the relative amounts of color needed. Users also need to communicate when they think they are close enough to check their answer.

Each color matching activity is timed to keep the students on task. Users are allowed at most 60 seconds to adjust the RGB values for each given color, although they may check sooner by clicking a button. Each response is given a score between 0 and 1000, where better scores are indicated by higher values. Scores are determined by the formula *score = max_distance- dist / max_distance * 1000* where *dist* is the distance between the target and users' RGB values, as determined by the position of the scroll bar, and *max_distance* is the maximum distance possible in the color space. The entire activity consists of matching ten colors. All scores are recorded in a ASCII log file.

3.3. The Chord Matcher

The Chord Matcher application is very similar to the Color Matcher, except that the users are charged with matching musical chords rather than RGB colors. The chords vary from a relatively easy major chord, to more difficult selections of three random notes of a 5-octave chromatic scale. Each user selects his or her note from the five octave range using a color coded scroll bar. As with the Color Matcher, users can only operate scroll bars of the same color as their mouse cursor.

Each user may hear his or her own note at will by clicking on a button. Similarly any user may audition the current team chord, or the target chord. Any user may also request a hint, up to a maximum of ten hints per session. The users are given 120 seconds to cor-

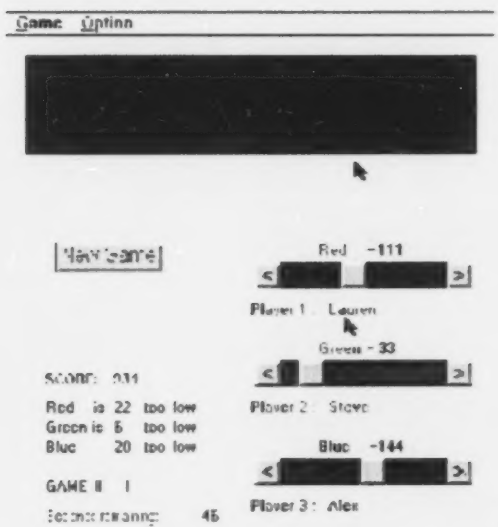


Figure 1. The Color Matcher User Interface.

rectly match the chord, although they may check sooner. The team's score is calculated as ð100 - 0.5 * (*dist_1 + dist_2 + dist_3)* û where *dist_i* is the distance each users note was from the target note, measured in half steps.

The user interface for the Chord Matcher is shown in Figure 2. As with the Color Matcher, effective play requires that students talk about the pitches in the chord and when to check their answer. Additionally users may negotiate over the use of shared resources, such the use of hints, and when the notes, team and target chords should be played.

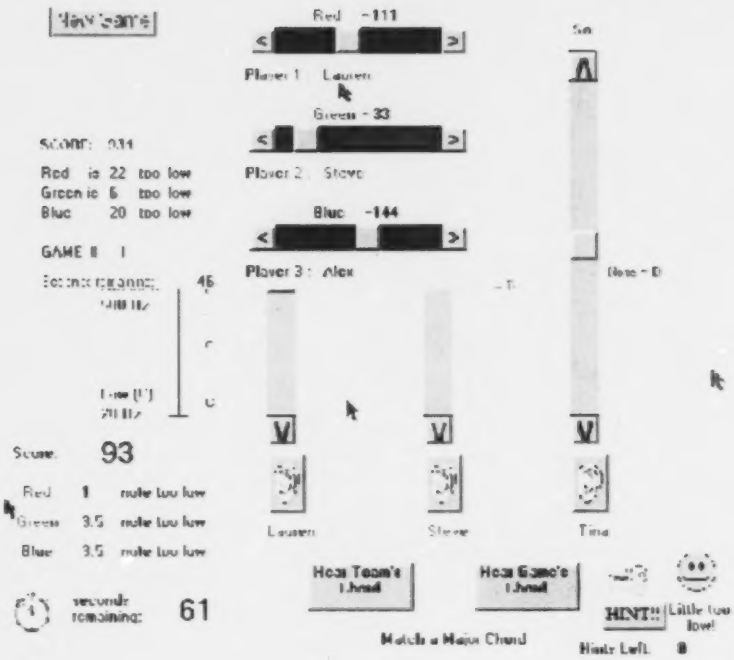


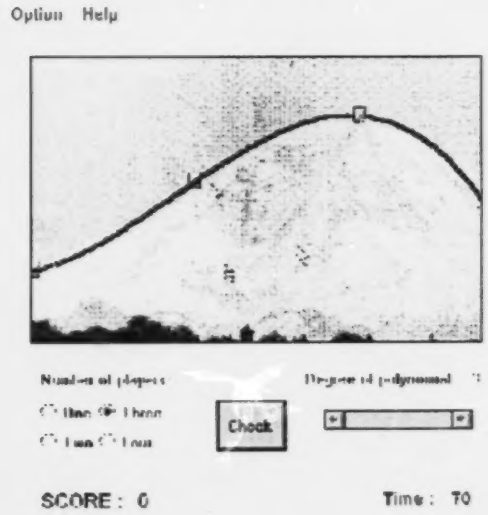
Figure 2: The Chord Matcher User Interface.

3.4. The Curve Fitter

The Curve Fitter activity is designed to allow students to explore how changing the location of control points modifies the shape of a polynomial curve. The user interface for the Curve Fitter is shown in Figure 3. This activity is designed for two to four users.

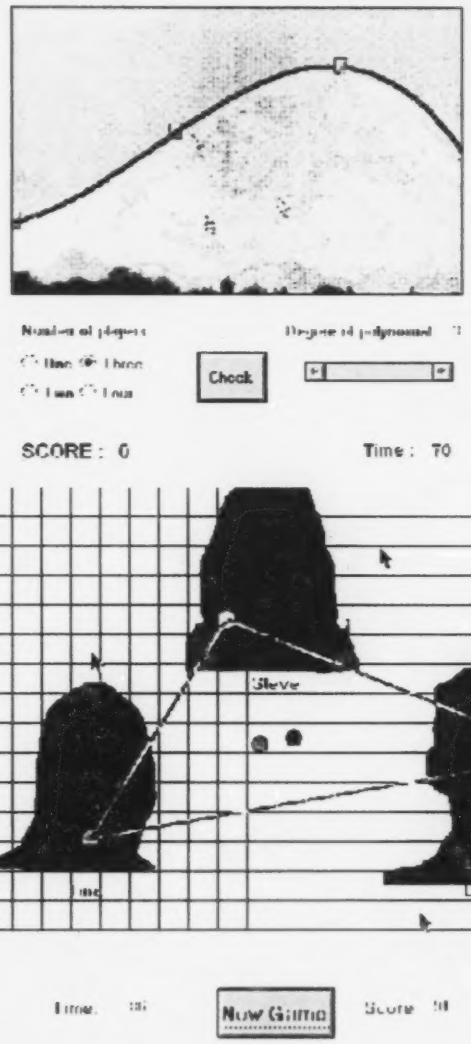
As with the Color Matcher and the Chord Matcher, each player controls a mouse which corresponds to a colored cursor on the screen. Each cursor can move a correspondingly colored control point on an image to modify the shape of the curve. The goal of the activity is to match a degree 1 to degree 5 polynomial with a curve in the image under 120 seconds. The users choose the degree of the polynomial before starting the activity. The scores range from 0 to 100 and is based on how close the curve is to the shape of the image. A high score is given for a close fit to the curve, thus a score of 100 represents the best fit of the curve.

Again the users need to communicate about the task they are performing. They typically discuss the shape of curve in relation to the locations of each of the control points. The only shared resource in this game is the button to check their answer.



BEST COPY AVAILABLE

Figure 3. The Curve Fitter User Interface.



mmunication about certain geometrical concepts. The activity may be played
in charge of the colored point matching his or her mouse cursor. If two users
oint of the line defined by their points with a predefined point

d in this activity, they must match the centroid of the triangle defined by their
· the movement of each of the users' points is limited to areas on the screen

nutes to complete their task. Since they can only move their own points, they
ı, the only shared screen resource is the button where the users can check
nages may be selected before each session.

*100 - (dist * factor)* where *dist* is the distance between the users' point
alue. Figure 4 shows the user interface for this activity.

Figure 4. The Midpoint Activity User Interface.

4. Preliminary User Feedback

4.1. Testing in 8th grade classes

We tested the Color Matcher in two 8th grade mathematics classes in a Seattle middle school. Our goal was not to conduct a scientific experiment, but rather to collect anecdotal evidence to support the use of the cooperative facilities provided by the MultiIn module.

The students were given a lecture of about 15 minutes about the RGB additive color system. The lecture was followed by a short pretest and questionnaire. The pretest was to assess their understanding of the concepts presented in the lecture and to access how much they might want to pursue further investigations of color.

We divided the classes into groups of six and brought each group to the PC lab. Three of each group used the *Multiple input device (MID)* Color Matcher described in Section 3.2. The other students used a *Single Input Device (SID)* version of the same software. The students were shown how to use the software, then left to do the activity by themselves. Student interactions were videotaped for later study. Finally the students completed a post-test to determine if their knowledge of the RGB color system had increased. Students were also had the opportunity to give their impressions of the program in the follow up questionnaire and on videotape.

The majority of the students responded that they had seen color mixing before, though it is uncertain if they meant the RGB

BEST COPY AVAILABLE

[redacted] irade schools. However, as a whole they had not previously used a computer to experiment with these concepts. The SID users were quiet, except to ask questions on how to operate the software. In contrast, the students using the MID software communicated a great deal, commanding each other to add "more green" or "less blue." Unlike the cooperative Logo studies done by Clements [3], there was very little conflict between these users as there was no contention for the input resources although some students did get slightly frustrated in discovering that they could not manipulate another player's color.

Overall the students enjoyed playing with both versions of the software. Some of the students restarted the activity after they had completed the test. One even used it to help answer the questionnaire. Another commented: "Having a computer at home you get used to working by yourself. But working together was more fun." Though another did say: "One player is better because its easier to get the right answer. You don't have to worry about what the other people are doing."

There was one group in particular that was very quiet and quite obviously did not get along very well. Their comments were mostly negative, though apparently aimed primarily at the teacher and at mathematics in general rather than the color matcher.

Scores on the test portion of the data are shown in Table 1. The students showed an increase in scores after using the color matcher, and there was slightly more of an increase among students that used the MID version of the software.

Table 1. Average scores on the pre- and post-test. Numbers in parentheses indicate sample size and *s* is the standard deviation.

	Single (7)	Multiple (15)
Pretest	2.33 s=1.66	4.13 s=2.80
Post-test	3.00 s=2.00	5.33 s=3.64
Change	28.76%	29.06%

4.2. Testing with college and graduate students

All of the collaborative applications were tested at the University of Washington. The volunteer testers, both graduate and undergraduate students most of whom had never seen this type of interface before, were selected in groups of three. Each group used each of the four applications, then answered a questionnaire in which they were asked to write down their impressions of the software and the interfaces. Additionally the group interactions were noted by three observers during the testing. No pre and post testing of knowledge was done.

In general, the users thought the interfaces were interesting and easy to use. Although all of the users gave some suggestions for improvement, none of the suggestions radically modified the underlying design for collaboration.

The students liked using the activities with partners, although they admitted they would rather use them by alone. (Perhaps this is due to the fact that they have all been taught to work and learn independently.) They did feel that the user interface made them communicate. One person noted, "[the interface] forced me to give or receive instructions." In fact, there seemed to be one person for each application who took the leading role in both explaining the underlying concepts of the activity and directing others. This type of tutor-learner relationship occurred most often during the testing of the Color Matcher and the Chord Matcher. In the case of the Color Matcher, there were a few people who didn't understand the RGB mixing scheme. This was even more evident in the Chord Matcher testing. In each group there was at least one person who was more versed in music theory than the rest. Between the tutors and the help screens, the groups decided that the best way to solve the easy (major) chord was to match the low note, then determine the middle and high notes by simply adding the proper amount of half-note steps. One person, who had perfect pitch and was able to match notes almost instantaneously, said "it's not fair for me to do this [activity]."

Unfortunately the users also pointed out that the Chord Matcher is inherently less collaborative than the others. One person's strategy was to "work on my own note individually, [then] offer suggestions to others after I'm done." The Color Matcher had the same problem, although the goal for the group is more tightly coupled than the Chord Matcher. In the former, the color homogeneously mix to form one, while in the latter the three notes are distinguishable by some people. Ultimately the team goal in the Color Matcher is the final color, whereas the goal for the Chord Matcher is to match the individual notes. In both of these applications, however, most users did not change their scroll bars while others were moving, presumably to see the effects of each scroll bar individually. One user commented they "picked something and let everyone else fight for a while, then made corrections."

The shared controls forced the users to communicate in a slightly different way. Typically a user asked the others in a group if he/she should or could use a shared control. In two distinct cases, users pressed a button without first asking for a consensus from the group, but did so only once. They realized afterwards that it was a shared resource and therefore its use should be based on a shared decision.

Finally, the users indicated that they did not feel hindered by the fact that they could only manipulate their own controls. In fact in the Curve Fitter and Midpoint activities, they felt that this forced them to communicate more. As one so aptly put it "You are forced to talk if you want to 'win' the game since you only control one piece of the puzzle."

5. Implementation Considerations

The four applications discussed in Section 3 were specifically designed as co-present CSCL activities for groups of 2-4 students sitting at one machine. The small group size was not only chosen to facilitate easy access to the machine, but also because groups of this size seem to work best for collaborative learning [5]. Furthermore, working in small collaborative groups teaches students skills needed in later life to interact with their peers in college or on the job.

The goals of these activities, as directed by the NCTM standards, are to favor conceptual learning over rote operations, emphasize practical uses of mathematics, encourage group discussions, and promote exploratory, open-ended discovery [13]. In particular, the programs described have relatively few shared screen objects (such as the check button). This forces the students to communicate with each other about manipulating the individual objects, rather than moving them themselves. In this way they are encouraged to share responsibility for the results of the activity. The use of multiple input devices for these activities not only avoids conflicts over hardware, but also has the added benefit that fewer machines are needed per classroom. The METIP/MultiIn environment has been developed on a 386/486/Pentium based PC platform running under Microsoft Windows because these systems are fairly prevalent in school systems. However, these stock machines only support standard single mouse/single keyboard hardware and operating systems. Therefore, we had to build support for simultaneous input from multiple students.

Video game joysticks are the most commonly used multiple user input devices. They have the advantage that they have wide support and can be plugged into a PC's standard game port. Unfortunately, joysticks are generally limited to two per machine and are difficult to use to control precise movement on screen. Using the keyboard to control cursor input has similar problems. While the keyboard is not expressly limited to one or even two users, it would be difficult to reasonably fit more than two people on any one keyboard at a time. The sole support of either of these types of controls would severely limit the range of activities MultiIn could support.

With the ability for relatively fine control of cursor movement, mice have been the standard graphical controllers for Windows based applications. Until recently there was no support for this type of multiple mouse interaction on one PC system. The ACCESS.bus "locator" communication protocol is designed to allow multiple input devices to operate on a single machine. However, the ACCESS.bus protocol requires a special board and therefore our activities cannot be used on a "vanilla" machine.

These types of devices are only a small subset of what is available. Given the plethora of multiple input solutions, we chose to design the MultiIn system to allow applications to treat all of these devices identically. Additionally, this design does not force the application developer commit to one particular type of hardware device.

Our technique for interacting with any input devices consists primarily of mapping specific device messages into our own standard format to eliminate variability. All that is required to integrate a device into the MultiIn system is a new module which specifies the translations from the device's messages into our own messages.

6. Conclusions and Future Work

To explore the potential for co-presence (not distance or teleconferencing-oriented) collaborative learning with computers, we have focused on activities in which several students simultaneously interact on a single screen. We have presented four co-present collaborative applications, the Color Matcher, the Chord Matcher, the Curve Fitter, and the Midpoint program. Each of these activities is designed to encourage exploration and discussion about the activity through interaction with the computer, as well as interaction with other students through verbal communication, gestures and body language. Through such natural, non-computer-mediated communication, students can utilize the language as well as the concepts of mathematics.

While not complete, the preliminary testing provides anecdotal evidence the hypothesis that students do tend to communicate effectively about the problem when using co-present multiple input applications. There was also, as anticipated, no contention for input resources. Further testing needs to be done to determine the effects of changing or restricting the ownership of objects on contention for resources, conflicts among users, communication and learning.

These activities have been built on a framework designed to allow the rapid development of input-device independent CSCL activities. This framework helps to hide one of the major stumbling blocks to the development of co-presence collaborative applications: the lack of standardization of multi-user input hardware.

The design of our multiple-user multiple-input interface is as yet incomplete. We are currently considering what types of user features are require to support collaborative learning interface applications and activities. With this knowledge we will extend MultiIn to include high level primitives for collaboratively manipulating objects. The primitives in this new system will support both co-present and distance cooperation. Thus this system will allow us to test the differences between these two forms collaboration.

Acknowledgments

We would like to thank Steve Fain and his students at the Meany Middle School for their cooperation in testing the multi-user and single-user color matching activities. We would also like to thank Leo Salemann for contributions to the design and

implementation of the color matching program. Finally, we gratefully acknowledge the partial support of the National Science Foundation under grant MDR-9155709, the Boeing Company, and the Washington Technology Center.

References

1. E. Bier and S. Freeman. MMM: A user interface architecture for shared editors on a single screen. In *UIST'91* , pages 79-86, 1991.

2. J. Campione, A. Brown, and M. Jay. Computers in a community of learners. In E. de Corte, M. C. Linn, H. Mandl, and L. Verschaffel, editors, *Computer-Based Learning Environments and Problem Solving* , pages 163-188. Springer-Verlag, 1992.

3. D. Clements and B. Nastasi. The role of social interaction in the development of higher-order thinking in Logo environments. In E. de Court, M. C. Linn, H. Mandl, and L. Verschaffel, editors, *Computer-Based Learning Environments and Problem Solving* , pages 229-247. Springer-Verlag, 1992.

4. N. Davidson. Introduction and overview. In N. Davidson, editor, *Cooperative Learning in Mathematics* , pages 1-20. Addison-Wesley, 1990.

5. D. Dockterman. Cooperative learning and computers. *SIGQUE Outlook* , 21(2):39-43, 1991.

6. M. Elart-Keys, D. Halonen, M. Horton, R. Kass, and P. Scott. User interface requirements for face to face groupware. in *CHI'90 Proceedings* , pages 295-301, 1990.

7. J. D. Foley, A. van Dam, S. K. Fiener, and J. F. Hughes. *Computer Graphics: Principles and Practice* , Addison-Wesley Publishing Co., Menlo Park, second edition, 1990.

8. W. W. Gaver, T. Moran, A. MacLean, L. Lovstrand, P. Dourish, K. Carter, and W. Buxton. Realizing a video environment: EuroPARC's RAVE system. In *CHI'92 Conference Proceedings* , pages 27-35, 1992.

9. H. Ishii, M. Kobayashi, and J. Grundin. Integration of inter-personal space and shared workspace: ClearBoard design and experiments. In *Proceedings of the CSCW* , pages 33-42, 1992.

10. J. J. Jehng, Y. Shih, S. Liang and T. Chan. TurtleGraph: A Computer Supported Cooperative Learning Environment. In *Proceedings of ED-MEDIA `94* , pages 293-298, 1994.

11. M. Muhlhauser and T. Rudenbusch. Cooperation support in computer-aided authoring and learning. In *Proceedings of ED-MEDIA `94* , pages 397-402, 1994.

12. B. K. Nastasi and D. H. Clements. Motivational and social outcomes of cooperative computer education environments. *Journal of Computing in Childhood Education* , 4(1):15-43, 1993.

13. National Council of Teachers of Mathematics. *Curriculum and Evaluation Standards for School Mathematics* . National Council of Teachers of Mathematics., Reston, Virginia, 1989.

14. National Research Council Board on Mathematical Sciences and Mathematical Sciences Education Board. *Education Counts: A Report to the Nation on the Future of Mathematics Education* . National Academy Press, Washington., D.C., 1989.

15. D. K. O'Neill and L. M. Gomez. The collaboratory notebook: a networked knowledge-building environment for project learning. In *Proceedings of ED-MEDIA `94* , pages 416-423, 1994.

16. M. Scardamalia and C. Bereiter. An architecture for collaborative knowledge building. In E. de Corte, M. C. Linn, H. Mandl, and L. Verschaffel, editors, *Computer-Based Learning Environments and Problem Solving* , pages 41-66. Springer-Verlag, 1992.

17. R. B. Smith. What you see is what I think you see. *SIGCUE Outlook* , 3(21):18-23, 1992.

18. M. Stefik, G. Foster, D. G. Bobrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Communications of the ACM* , 30(1):32-47, 1987.

18. C. G. Wolf and J. R. Rhune. Communication and information retrieval with a pen-based meeting support tool. In *Proceedings of the CSCW* , pages 322-328, 1992.

Authors' Addresses

Lauren J. Bricker, Steven L. Tanimoto, Alex I. Rothenberg, Danny C. Hutama, and Tina H. Wong : Department of Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98195-2350. {bricker, tanimoto} @cs.washington.edu.